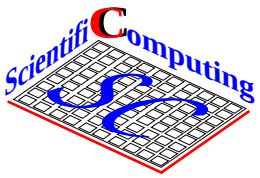

Strong Coupling Methods

Hermann G. Matthies, Jan Steindorf
Institute of Scientific Computing
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-06

September 2002

Strong Coupling Methods

Hermann G. Matthies, Jan Steindorf
Department of Computer Science
Technical University Braunschweig
Brunswick, Germany

Informatikbericht Nr.: 2002-06

September 2002

Contributed to the Conference Proceedings

Revised and expanded version of a contribution to the Stuttgart Multifield Conference, Stuttgart, Germany, April 2002

Location
Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Strasse 65
D-38106 Braunschweig

Postal Address
Institut für Wissenschaftliches Rechnen
Technische Universtität Braunschweig
D-38092 Braunschweig
Germany

Contact
Phone: +49-(0)531-391-3000
Fax: +49-(0)531-391-3003
E–Mail: wire@tu-bs.de

Strong Coupling Methods

Hermann G. Matthies Jan Steindorf
Institute of Scientific Computing
Technische Universität Braunschweig
Brunswick, Germany
September 2002

Abstract

We consider the procedure of computing the response of a coupled problem with a partitioned approach. Often we have existing procedures or even software to solve each sub-problem separately, and want to couple both. This setting seems to allow only the so-called weak coupling which is not sufficient for some problems. The so-called strong coupling — a totally implicit formulation — requires iteration in each time step. With the partitioned approach, one simple computational procedure is similar to a block-Gauss-Seidel iteration. We show why this approach may experience difficulties, and how they may be circumvented with block-Newton methods, still in the partitioned framework, by only using the solvers for the two sub-problems. We supply some examples from fluid-structure coupling.

Keywords: partitioned methods, strong coupling, block-Newton methods, fluid-structure-interaction

Contents

1 Introduction

Our interest in this field started from fluid-structure interaction (FSI) problems [1]. These often show strong interplay between the fluid and the structure [1, 2, 3, 4, 5, 6, 7], e.g. in the design of aircraft [8, 9] and in many other situations [10].

But apart from FSI-problems, the algorithms may often be used in many other situations where coupled problems arise [11], e.g. thermo-mechanical [12] coupling, soil-pore fluid interaction [13], to name but a few.

There are different ways to tackle a coupled interaction problem, cf. [14, 15, 16, 17]. One possibility is to develop new software and solution methods for each of these coupled applications, as undoubtably will happen in some areas. This is referred to as a *monolithical* approach [18], or sometimes as the *direct* method [19]. On the other hand, we shall assume here that the methods and software systems which have been developed for either applications will continue to be used, in our example fluid or structural software. Thefore we consider *partitioned* methods [20, 21, 14, 6, 22, 23] — also known as *iterative* methods [19] for fluid-structure interaction, i.e. separate solvers are used for the fluid and the structure [24, 25].

For stability reasons, often a fully implicit formulation has to be used [7, 19]. In this approach, we have to solve a large system of nonlinear equations with the use of the (iterative) solvers for the subsystems. Commonly this is performed with block-Jacobi, block-Gauss-Seidel or related relaxation methods [26]. These simple methods do not always converge [14, 27, 25, 28, 29, 30, 31]. We will introduce here a superior approximative block-Newton method [32, 25, 28, 29, 30, 31].

These ideas arise and can be applied in a multitude of other areas such as general multifield/multiphysics phenomena [14, 9], co-simulation and multi-numerics [33, 34, 27, 35], domain-decomposition [36], wave-form relaxation [37], and are in line with efforts of parallelisation and modularisation of software. Algorithms derived from Newton's method have been considered [38, 39]; here we have to observe the constraint that certain parts of the Jacobian system during the implicit iteration may not be accessible. In the end not the — incremental — equlibrium equation are solved, but an equivalent, numerically better conditioned system [32, 25, 28, 29, 30, 31].

2 Coupled Systems

For the sake of simplicity we shall only consider a global system composed of two subsystems. It is no problem to extend the ideas and methods presented here to more than two coupled subsystems, but this makes harder to convey the basic

ideas, which may be seen already in the simplest case of two subsystems.

Often the coupling will introduce additional variables — we shall show this later for the FSI example — and hence algebraic equations in addition to the subsystem equations.

2.1 Pure Differential Coupling

Let us assume that the time evolution of the first subsystem is given by a differential equation of the form

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2) \ , \tag{1}$$

where $\boldsymbol{x}_1 \in \mathcal{X}_1$, $\boldsymbol{x}_2 \in \mathcal{X}_2$ are elements of some Banach-spaces, and the superimposed dot denotes the time derivative. The spaces $\mathcal{X}_1$ and $\mathcal{X}_2$ may be spaces of functions, and $\boldsymbol{f}_1$ a differential operator, so that Eq.(1) may describe subsystems modeled by partial differential equations, such as fluid flow. The variables $\boldsymbol{x}_2$ are not further specified in this model, they provide the coupling to the second subsystem, which is described completely analogous to Eq.(1):

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2, \boldsymbol{x}_1) \ . \tag{2}$$

Here analogous remarks as with Eq.(1) apply. The coupled system Eqs.(1,2) is nothing but a larger evolution system for the combined vector $(\boldsymbol{x}_1, \boldsymbol{x}_2)^T$, and actually may come from splitting a larger system into smaller ones for parallelisation, such as in wave-form relaxation [37]. For all variables we have differential evolution equations, and therefore we refer to this case as pure differential coupling. In a later section we shall describe the case when the coupling is modeled by "algebraic" equations, and the subsystems themselves may be differential algebraic equations (DAEs).

2.1.1 Explicit Coupling

We may assume that both subsystems Eqs.(1,2) have been discretised — if necessary both in space and — in time with some appropriate method. If the methods for the individual subsystems are both explicit, probably the most natural is to stay explicit also for the coupled system. For the sake of simplicity of exposition we shall assume here that both subsystems have the same time step Δt, otherwise what we say refers to their synchronisation points, and one subsystem may be performing what is called sub-cycling. Let us denote the the approximation to the solution at step n by $\boldsymbol{x}_j^{(n)}, \ (j = 1, 2)$, then the explicit integration algorithms — assumed to be single step for ease of notation — for the two subsystems may be

denoted by

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\varphi}_1(\boldsymbol{x}_1^{(n-1)}, \boldsymbol{y}_2) \ , \tag{3}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\varphi}_2(\boldsymbol{x}_2^{(n-1)}, \boldsymbol{y}_1) \ , \tag{4}$$

where the two functions $\boldsymbol{y}_j(t),\ (j = 1, 2)$ are assumed given, and dependence on time t or the time step Δt has been suppressed. They are to represent the variables of the other subsystem in the time interval of length Δt from t_{n-1} to t_n, but of course these are not known yet! Staying explicit we may extrapolate these by some function $\Psi_j(\boldsymbol{x}_j^{(n-1)})$, $(j = 1, 2)$ of the past values of the approximate solution, in the simplest case one may assume the value at t_{n-1} as constant throughout the interval, i.e. $\boldsymbol{y}_j(t) = \Psi_j(\boldsymbol{x}_j^{(n-1)}) \equiv \boldsymbol{x}_j^{(n-1)}$, $(j = 1, 2)$. This leads to the simplest case of *weak* or *loose* coupling, the so-called *staggering* method [22, 40], here with explicit subsystem integrators:

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\varphi}_1(\boldsymbol{x}_1^{(n-1)}, \Psi_2(\boldsymbol{x}_2^{(n-1)})) = \boldsymbol{\varphi}_1(\boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n-1)}) \ , \tag{5}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\varphi}_2(\boldsymbol{x}_2^{(n-1)}, \Psi_1(\boldsymbol{x}_1^{(n-1)})) = \boldsymbol{\varphi}_2(\boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n-1)}) \ . \tag{6}$$

With this, the method is completely specified. The single system integrators are most likely going to have an associated critical time step, and by coupling the systems in this explicit fashion the critical time step for the subsystem will most likely decrease. The two sub-steps in Eqs.(5,6) can be performed in parallel. If we sacrifice this inherent parallelism, we can formulate a partly implicit method: First perform Eq.(5), to be followed by

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\varphi}_2(\boldsymbol{x}_2^{(n-1)}, \Psi_1(\boldsymbol{x}_1^{(n-1)})) = \boldsymbol{\varphi}_2(\boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n)}) \ , \tag{7}$$

where the new value $\boldsymbol{x}_1^{(n)} = \Psi_1(\boldsymbol{x}_1^{(n-1)})$ — we may say that we have as extrapolation $\Psi_1(\boldsymbol{x}_1^{(n-1)}) = \boldsymbol{\varphi}_1(\boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n-1)})$ — is already used. This is how the staggering method is mostly applied. Of course the combined method is still explicit, but only "half as much" as before.

In case that the subsystem integrators are implicit, e.g. in order not to be constrained by a critical time step in the subsystem, the new approximations $\boldsymbol{x}_j^{(n)},\ (j = 1, 2)$ appear also on the r.h.s. of Eqs.(8,9), meaning that we have to solve the equation system:

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \boldsymbol{y}_2) \ , \tag{8}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \boldsymbol{y}_1) \ . \tag{9}$$

Here again the two functions $\boldsymbol{y}_j(t),\ (j = 1, 2)$ are assumed to be given, and as before we need two extrapolation operators

$$\boldsymbol{y}_j = \Psi_j(\boldsymbol{x}_j^{(n-1)}) \ , \quad (j = 1, 2) \ . \tag{10}$$

Applying the same simple approximation

$$\Psi_j(\boldsymbol{x}_j^{(n-1)}) \equiv \boldsymbol{x}_j^{(n-1)} \ , \quad (j = 1, 2) \tag{11}$$

as in Eqs.(5,6), we arrive at

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \Psi_2(\boldsymbol{x}_2^{(n-1)})) = \boldsymbol{\phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n-1)}) \ , \tag{12}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \Psi_1(\boldsymbol{x}_1^{(n-1)})) = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n-1)}) \ . \tag{13}$$

This is a staggering type algorithm for implicit integrators. The two subsystems Eqs.(12,13) may be solved independently of each other, but the method, considered for the system as a whole, is explicit, and will again introduce limitations connected with a critical time step associated to the global system. To circumvent this, generally we have to introduce a global coupling, although there are methods to retain at least the linear stability characteristics in the staggered approach [21, 14, 41]. We should mention that naturally the partly implicit variant of Eq.(6) is also possible: After performing Eq.(12), replace Eq.(13) by

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \Psi_1(\boldsymbol{x}_1^{(n-1)})) = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n)}) \ ; \tag{14}$$

where again we may say that we have as extrapolation $\Psi_1(\boldsymbol{x}_1^{(n-1)}) = \boldsymbol{\varphi}_1(\boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n-1)})$.

2.1.2 Implicit Coupling

To alleviate the time step restriction still present in Eqs.(12,13), we may formulate the whole system in an implicit way, extrapolating $\boldsymbol{y}_j(t) = \Psi_j(\boldsymbol{x}_j^{(n-1)}) \equiv \boldsymbol{x}_j^{(n)}, \ (j = 1, 2)$ with the constant — and yet unknown — value of the approximate solution at the end of the time step. Again here some more elaborate approximation is possible, but we restrict ourselves to this simplest and most used case for the sake of brevity. The equation system to be solved at each time step now reads

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \Psi_2(\boldsymbol{x}_2^{(n-1)})) = \boldsymbol{\phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n)}) \ , \tag{15}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \Psi_1(\boldsymbol{x}_1^{(n-1)})) = \boldsymbol{\phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n)}) \ . \tag{16}$$

This is a case of *strong* or *tight* coupling, and the results from this are completely equivalent to what would be achieved by a monolithical formulation, although we arrived here through a partitioned approach. The equations Eqs.(15,16) are a coupled system of equations, and unlike Eqs.(12,13) can not be solved independently of each other. Hence, even if we have methods to solve each of

Eqs.(12,13) by itself, this does not appear to help us in the solution of the global system, we need a global iteration. But in section 5 we shall see that the methods for the subsystems can be put to good use in the solution of the global system.

Thus although the formulation of the *partitioned* but *strongly coupled* approach is equivalent to a *monolithical* approach, the methods to solve the coupled system will be different: In the monolithical approach we have all the information desired about the subsystems at our disposal, and the solution methods are sometimes termed as *direct* [19], although the actual solution process may — and in many cases will — very well be iterative.

Starting from the partitioned approach, we do not have all the information desired about the subsystems at our disposal, and there has to be some kind of iteration across the subsystems. Therefore this approach is sometimes termed the *iterative* [19] method. The iteration across the subsystems means also that information has to be exchanged, and that, if the subsystems are solved iteratively, we have a nested iteration. This always brings up the question how these iteration processes can be coordinated. In section 5 we will show in what sense the global iteration can control the inner iteration of the individual iterative subsystem solvers.

2.2 Differential and Algebraic Coupling

In the last paragraph we arrived at Eqs.(15,16), a global system of equations to be solved at each time step. It will turn out that the case of two coupled differential algebraic equations (DAEs) is no more difficult conceptually at each time step, so we shall look at this situation next [27].

Assume that the two subsystems are DAEs of index 1, denoted by

$$\dot{\boldsymbol{x}}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{z}) \;, \tag{17}$$

$$\mathbf{0} = \boldsymbol{g}_1(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{z}) \;. \tag{18}$$

The description is similar to Eq.(1), only that we have additionally a local — to the first subsystem — algebraic variable $\boldsymbol{y}_1$ and a global one $\boldsymbol{z}$. As with the evolution law $\boldsymbol{f}_1$ in Eq.(17), also the function $\boldsymbol{g}_1$ in the "algebraic" equation Eq.(18) may be a spatial differential operator. Completely analogous, just by switching indices, we model the second subsystem by

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_2(\boldsymbol{x}_2, \boldsymbol{x}_1, \boldsymbol{y}_2, \boldsymbol{z}) \;, \tag{19}$$

$$\mathbf{0} = \boldsymbol{g}_2(\boldsymbol{x}_2, \boldsymbol{x}_1, \boldsymbol{y}_2, \boldsymbol{z}) \;. \tag{20}$$

The two subsystems are now not only coupled by the differential variables $\boldsymbol{x}_j,\ (j = 1, 2)$ as before, but we additionally allow for an algebraic coupling

$$\mathbf{0} = \boldsymbol{h}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y}_1, \boldsymbol{y}_2, \boldsymbol{z}) \;, \tag{21}$$

through the global algebraic variable $\boldsymbol{z}$.

As already mentioned, we assume that each subsystem is a DAE of index 1, and that the global system is a DAE of index 1. This means that the matrices

$$D_{y_j}\boldsymbol{g}_j\ , \qquad \begin{bmatrix} D_{y_j}\boldsymbol{g}_j & D_z\boldsymbol{g}_j \\ D_{y_j}\boldsymbol{h} & D_z\boldsymbol{h} \end{bmatrix},\ (j=1,2)\ , \qquad \begin{bmatrix} D_y\boldsymbol{g} & D_z\boldsymbol{g} \\ D_y\boldsymbol{h} & D_z\boldsymbol{h} \end{bmatrix} \tag{22}$$

have to be regular, where D_q is the partial derivative w.r.t. q, and we have set $\boldsymbol{g} = (\boldsymbol{g}_1, \boldsymbol{g}_2)^T$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2)^T$.

Upon discretising equations Eqs.(17,18) and Eqs.(19,20) in time with an implicit method, we will again arrive at an implicit equation to be solved for each subsystem. Treating also the differential variable from the other subsystem implicitly like in Eqs.(15,16), and also treating the global algebraic condition Eq.(21) implicitly, we arrive at a global system to be solved:

$$\boldsymbol{x}_1^{(n)} = \boldsymbol{\Phi}_1(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_1^{(n-1)}, \boldsymbol{x}_2^{(n)}, \boldsymbol{y}_1^{(n)}, \boldsymbol{z}^{(n)})\ , \tag{23}$$

$$\boldsymbol{x}_2^{(n)} = \boldsymbol{\Phi}_2(\boldsymbol{x}_2^{(n)}, \boldsymbol{x}_2^{(n-1)}, \boldsymbol{x}_1^{(n)}, \boldsymbol{y}_2^{(n)}, \boldsymbol{z}^{(n)})\ , \tag{24}$$

$$\boldsymbol{0} = \boldsymbol{H}(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_2^{(n)}, \boldsymbol{y}_1^{(n)}, \boldsymbol{y}_2^{(n)}, \boldsymbol{z}^{(n)})\ . \tag{25}$$

Again this is a coupled system of equations, but we may only have solvers for each subsystem separately.

3 Fluid-Structure Interaction

Before describing in the abstract setting of the preceeding section 2 how to solve these systems numerically by using the subsystem solvers, we shall take a glance at the FSI problem, and how to formulate it such that it fits the above setting.

3.1 The Fluid

We assume the fluid to be modeled adequately as an incompressible Newtonian fluid, satisfying the appropriate Navier-Stokes equation. As we want to couple this with a moving structure, we have to allow for a moving boundary. We take account of this by formulating the Navier-Stokes equation in an Arbitrary Lagrangean-Eulerian (ALE) [4, 42, 43] framework in the moving spatial fluid domain Ω_f:

$$\varrho_f\left(\dot{v} + ((v-\dot{\chi})\cdot\nabla)v\right) - \operatorname{div}\sigma + \nabla p = r_f, \tag{26}$$

$$2\sigma = \nu(\nabla v + (\nabla v)^T), \qquad \operatorname{div} v = 0. \tag{27}$$

The boundary $\partial\Omega_f$ we assume to be divided into three disjoint parts $\partial\Omega_f = \Gamma_v \cup \Gamma_q \cup \Gamma_c$ where on Γ_v the velocity is prescribed, on Γ_q the traction is given, and Γ_c

is the coupling boundary, where the coupling conditions will be specified below. Here ϱ_f is the fluid density, v the velocity, σ the viscous stress, χ is the position of the reference ALE-coordinate system and $\dot{\chi}$ its velocity. The fluid shear viscosity is denoted by ν, p is the pressure, r_f the body force in the fluid, and the differential operators are in the spatial frame.

There is no essential difficulty in assuming other fluid models, such as compressible or non-Newtonian ones, the abstract formulation introduced earlier in section 2 would still be fitting. But it is the example of an incompressible fluid which makes it more challenging in one respect, as the incompressibility condition is an "algebraic" constraint.

The movement of the ALE-coordinate system $\chi(t)$ is not specified yet, this will be done after we have considered the discretisation.

3.2 The Structure

The structure we assume to be modeled by a neo-Hookean material, hence the equilibrium equation takes the following form in a Lagrangean framework in a fixed material domain Ω_s:

$$\varrho_s \ddot{u} - \text{DIV}\,(FS) = r_s, \qquad F = \text{GRAD}\, u \tag{28}$$

$$S = \lambda(\text{tr}\, E)I + 2\mu E, \qquad 2E = (C - I), \qquad C = F^T F \tag{29}$$

The boundary $\partial\Omega_s$ we again assume divided into three disjoint parts $\partial\Omega_s = \Gamma_u \cup \Gamma_t \cup \Gamma_c$, where on Γ_u the displacements u are prescribed, on Γ_t the tractions, and Γ_c is the coupling boundary with the fluid. The other quantities are the structure density ϱ_s, the displacement gradient F, the second Piola-Kirchhoff stress S, and the body load r_s. The Lamé moduli are denoted by λ and μ, and E is the Lagrange-Green strain, derived from the Cauchy-Green tensor C, and the differential operators are in the material frame.

Again here it would be possible to assume other models for the solid, e.g. non-elastic ones with internal variables such as plasticity or damage, they would still fit into the general setting introduced in section 2.

3.3 The Interface

On the coupling boundary Γ_c let a unique normal n be defined in the spatial frame. Then the coupling conditions may be expressed as requiring the velocities to coincide at the location $\chi(t) = \chi_0 + u(\chi_0, t)$ and the tractions from fluid and structure to balance each other:

$$v(\chi(t), t) = \dot{u}(\chi_0, t), \tag{30}$$

$$(\sigma - pI) \cdot n = -\frac{1}{J} FSF^T \cdot n\,, \qquad J = \det F\,. \tag{31}$$

4 FSI as a Coupled DAE

We shall further assume that the fluid — described by Eqs.(26,27), the structure — described by Eqs.(28,29), and the interface — described by Eqs.(30,31), have been discretised; each one may use her favourite discretisation, e.g. finite elements, finite volumes etc., e.g. [11, 44, 45].

4.1 Discrete Form of FSI

Up to now we have only hinted at how to move the reference coordinate system in the fluid domain. Several possibilities exist [46, 47, 48, 7], here we have followed [49] and modeled the connections between the nodes in the fluid domain as elastic springs. This fictitious inertia-free elastic body has displacement loading from the moving structure, on the outer boundaries it is fixed. The traction balance Eq.(31) introduces additional forces on the fluid, and the reaction force on the solid. Hence for the fluid the complete discrete equations of motion are [25]:

$$\boldsymbol{M}_f\dot{\boldsymbol{v}} + \boldsymbol{N}(\boldsymbol{v}-\dot{\boldsymbol{\chi}})\boldsymbol{v} + \boldsymbol{K}_f\boldsymbol{v} + \boldsymbol{B}_f\boldsymbol{p} = \boldsymbol{r}_f + \boldsymbol{T}_f^T\boldsymbol{\tau}\ , \tag{32}$$

$$\boldsymbol{B}_f^T\boldsymbol{v} = \boldsymbol{0}\ , \tag{33}$$

$$\boldsymbol{K}_g\boldsymbol{\chi} = \boldsymbol{A}\boldsymbol{u}\ . \tag{34}$$

The terms in Eq.(32) are the discrete analogues of those in Eq.(26), the term $\boldsymbol{r}_f$ includes the prescribed boundary stresses, and the additional term $\boldsymbol{T}_f^T\boldsymbol{\tau}$ comes from the interaction with the structure, where $\boldsymbol{\tau}$ is the stress on the fluid-structure-interface. In Eq.(33) we may recognise the discrete form of the incompressibility condition, and Eq.(34) describes the movement $\boldsymbol{\chi}(t)$ of the fluid domain, driven by the structure displacements $\boldsymbol{u}$.

In a similar vein we obtain for the structure

$$\boldsymbol{M}_s\ddot{\boldsymbol{u}} + \boldsymbol{K}_s(\boldsymbol{u})\boldsymbol{u} = \boldsymbol{r}_s - \boldsymbol{T}_s^T\boldsymbol{\tau}\ . \tag{35}$$

To this we add the discrete coupling condition for the velocities,

$$\boldsymbol{T}_f\boldsymbol{v} = \boldsymbol{T}_s\dot{\boldsymbol{u}}\ . \tag{36}$$

This combined set, Eqs. Eqs.(32,33,34) and Eqs.(35,36), is a system of differential algebraic equations (DAEs) in the time variable t of index 2, as can be easily verified. In order to allow an easier numerical treatment and to make them fit the general formulation in subsection 2.2, they will be converted to index 1 by differentiation [25].

There are quite a few items we have glossed over in our quick description, as these are not central to our issue. Nevertheless, for a proper implementation of

FSI they have to be dealt with. This includes, but is not limited to, the problem of non-matching grids for fluid, structure, and possibly interface, and the ALE-fluid domain. There are different possibilities to tackle those problems, such as consistent interpolation and mortar elements, to name but a few [50, 51, 52, 53]. Another problem is a kind of consistency when moving the ALE-fluid mesh; the numerical time stepping algorithm together with the spatial discretisation has to satisfy a special property, referred to as the geometric conservation law (GCL) [46, 54].

4.2 Index and Order Reduction

First we will reduce the index of the fluid system by differentiation. Differentiating Eq.(33) once gives $\boldsymbol{B}_f^T \dot{\boldsymbol{v}} = \boldsymbol{0}$. As $\boldsymbol{M}_f$ is non-singular, we may solve for $\dot{\boldsymbol{v}}$ from Eq.(32) and insert this into the above relation, giving

$$\boldsymbol{B}_f^T \boldsymbol{M}_f^{-1}(\boldsymbol{r}_f + \boldsymbol{T}_f^T \boldsymbol{\tau} - \boldsymbol{N}_f(\boldsymbol{v} - \boldsymbol{\psi})\boldsymbol{v} - \boldsymbol{K}_f \boldsymbol{v} - \boldsymbol{B}_f \boldsymbol{p}) = \boldsymbol{0} \ , \tag{37}$$

where we have set $\boldsymbol{\psi} := \dot{\boldsymbol{\chi}}$ for the grid velocity. In the same manner we differentiate equation Eq.(34) for the grid movement

$$\boldsymbol{K}_g \boldsymbol{\psi} - \boldsymbol{A}\boldsymbol{w} = \boldsymbol{0} \ , \tag{38}$$

where we have at the same time introduced the new variable $\boldsymbol{w} := \dot{\boldsymbol{u}}$, the structural velocities.

With this relation the structural equation Eq.(35) can be reduced to first order in a standard manner:

$$\boldsymbol{M}_s \dot{\boldsymbol{w}} + \boldsymbol{K}_s(\boldsymbol{u})\boldsymbol{u} = \boldsymbol{r}_s - \boldsymbol{T}_s^T \boldsymbol{\tau} \ . \tag{39}$$

4.3 The DAE Correspondence

For our concrete application, the abstract Eqs.(17,18) may now be identified with the following terms [25]:

$$\boldsymbol{x}_1 = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\chi} \end{bmatrix} \ , \quad \boldsymbol{y}_1 = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{p} \\ \boldsymbol{\psi} \end{bmatrix} \ , \quad \boldsymbol{f}_1 = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\psi} \end{bmatrix} \ , \quad \boldsymbol{z} = \boldsymbol{\tau} \ , \tag{40}$$

$$\boldsymbol{g}_1 = \begin{bmatrix} \boldsymbol{M}_f \boldsymbol{b} + \boldsymbol{N}_f(\boldsymbol{v} - \boldsymbol{\psi})\boldsymbol{v} + \boldsymbol{K}_f \boldsymbol{v} + \boldsymbol{B}_f \boldsymbol{p} - \boldsymbol{r}_f - \boldsymbol{T}_f^T \boldsymbol{\tau} \\ -\boldsymbol{B}_f^T \boldsymbol{M}_f^{-1}(-\boldsymbol{N}_f(\boldsymbol{v} - \boldsymbol{\psi})\boldsymbol{v} - \boldsymbol{B}_f \boldsymbol{p} - \boldsymbol{K}_f \boldsymbol{v} + \boldsymbol{r}_f + \boldsymbol{T}_f^T \boldsymbol{\tau}) \\ \boldsymbol{K}_g \boldsymbol{\psi} - \boldsymbol{A}\boldsymbol{w} \end{bmatrix} \ , \tag{41}$$

so the first subsystem is the fluid with the ALE-grid. In Eq.(40,2) we have introduced the new variable $\boldsymbol{b} := \dot{\boldsymbol{v}}$, the fluid acceleration.

The second subsystem is the structure,

$$\boldsymbol{x}_2 = \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{bmatrix}, \quad \boldsymbol{y}_2 = \boldsymbol{a}, \quad \boldsymbol{f}_2 = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{a} \end{bmatrix}, \quad \boldsymbol{z} = \boldsymbol{\tau} \tag{42}$$

$$\boldsymbol{g}_2 = \boldsymbol{M}_s\boldsymbol{a} + \boldsymbol{K}_s(\boldsymbol{u})\boldsymbol{u} - \boldsymbol{r}_s + \boldsymbol{T}_s^T\boldsymbol{\tau};, \tag{43}$$

where no index reduction, but only a standard order reduction of the differential equation was needed. In Eq.(42,2) we have introduced the new variable $\boldsymbol{a} := \dot{\boldsymbol{w}} = \ddot{\boldsymbol{u}}$, the structural acceleration.

The final item to identify is the global coupling condition,

$$\boldsymbol{h} = \boldsymbol{T}_f\boldsymbol{b} - \boldsymbol{T}_s\boldsymbol{a}\ , \tag{44}$$

which expresses the equality of the accelerations at the fluid-structure interface.

Let us remark that in the chosen formulation the differential equation only gives the definition of the various variables, whereas the equilibrium equations have gone into the algebraic conditions. What is still left is to verify the index 1 conditions Eq.(22). This is a lengthy calculation [25], but the index 1 conditions are indeed satisfied as long as $\boldsymbol{K}_g$, $\boldsymbol{M}_f$, and $\boldsymbol{M}_s$ are regular — in our case they even can be assumed to be symmetric positive definite.

5 Numerical Procedures for Partitioned Methods

In section 2 we have already tried to give an overview on how this might be possible. We also see that the computational procedures are very dependent on the formulation chosen for the global system. If, as we have done here, the coupled system is formulated as a differential-algebraic equation, we are almost inevitably led to an at least partly implicit formulation, and such mixed time-discretisations have also been investigated [55, 56]. For our example this problem already comes within one subsystem, namely the fluid. Here we have an incompressibility constraint, and practically all discretisations have some implicit elements [45].

For our problem we now have

- *the fluid problem* with the components
    1. the Navier-Stokes equation Eq.(26) — a partial differential equation in space and time, or its semi-discrete form Eq.(32), an ordinary differential equation in time.
    2. the incompressibility constraint Eq.(27) — a partial differential equation in space, or its discrete form Eq.(33), an algebraic constraint.

3. the ALE-grid movement, here only formulated in discrete form as an additional algebraic constraint Eq.(34).

- *the structure problem*, a partial differential equation in space and time, and of second order in time Eq.(28), or in its semi-discrete form Eq.(35) a second order ordinary differential equation in time.

- *the coupling condition*, a partial differential equation Eqs.(30,31) on the spatial interface between the fluid and the structural domain, or in its discrete form Eq.(36) an algebraic constraint.

The grid movement could very well be taken as a subsystem in its own right, but because it is so intimately connected to the fluid problem, it is customarily treated simultaneously.

Through index and order reduction the above problems have been reduced to the abstract coupled DAE system Eqs.(40–43). We assume that at least — as it is otherwise trivial — both subsystems have some implicit parts; something which is true for our example and also in line with common numerical wisdom for the discretisation of DAEs. In our partitioned and modular approach this also means that we have solvers for each subsystem, we write them here in fixed point form, as they may be used during the iterative solution process for each subsystem. This certainly includes also direct solvers, they may be simply regarded as very fast convergent iterative solvers — they only need one iteration.

After discretisation in time, we then arrive at the concrete versions of our abstract Eqs.(23-25). The first Eq.(23) solves for the variables $(\boldsymbol{x}_1, \boldsymbol{y}_1)^T = (\boldsymbol{v}, \boldsymbol{\chi}, \boldsymbol{b}, \boldsymbol{p}, \boldsymbol{\psi})^T$ assuming the others are given, the second Eq.(24) solves for $(\boldsymbol{x}_2, \boldsymbol{y}_2)^T = (\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{a})^T$ assuming the others are given, and the last Eq.(25) solves for $\boldsymbol{z} = \boldsymbol{\tau}$ assuming the others are given. For simplicity, and as it is also customarily done that way, when solving we shall include the discrete coupling condition with one of the other two subsystems, i.e. either with the fluid, meaning that on the coupling interface the velocities are prescribed (Dirichlet data), and the interface tractions are passed from the fluid to the structure (Neumann data); or the other way around. In the first case we shall jointly denote the variables as $\boldsymbol{\xi} := (\boldsymbol{x}_1, \boldsymbol{y}_1, \boldsymbol{z})^T = (\boldsymbol{v}, \boldsymbol{\chi}, \boldsymbol{b}, \boldsymbol{p}, \boldsymbol{\psi}, \boldsymbol{\tau})^T$ and $\boldsymbol{\zeta} := (\boldsymbol{x}_2, \boldsymbol{y}_2)^T = (\boldsymbol{u}, \boldsymbol{w}, \boldsymbol{a})^T$, and in the second case the variable $\boldsymbol{z} = \boldsymbol{\tau}$ is included in $\boldsymbol{\zeta}$ and not in $\boldsymbol{\xi}$.

As we want now to concentrate on how to solve the coupled Eqs.(23–25), we shall drop the time step counter, and also suppress the dependence on the previous time steps. We may assume both iterative solution processes are in fixed point form, i.e. the iteration

$$\boldsymbol{\xi}_\kappa = \boldsymbol{F}_1(\boldsymbol{\xi}_{\kappa-1}, \boldsymbol{\zeta}) \ , \quad \kappa = 1, 2, \dots \ ; \tag{45}$$

converges for reasonable starting values $\boldsymbol{\xi}_0$ with given $\boldsymbol{\zeta}$. Similarly, the iteration process

$$\boldsymbol{\zeta}_\kappa = \boldsymbol{F}_2(\boldsymbol{\zeta}_{\kappa-1}, \boldsymbol{\xi}) \ , \quad \kappa = 1, 2, \dots \ ; \tag{46}$$

converges for reasonable starting values $\boldsymbol{\zeta}_0$ with given $\boldsymbol{\xi}$.

5.1 Non-linear Block-Jacobi

The conceptually simplest way to use these two solvers F_1 and F_2 to solve the combined system is a non-linear block-Jacobi process, or in the context of domain decomposition it would be called an additive or parallel Schwarz procedure [36]. Given $\boldsymbol{\xi}_{\kappa-1}$ and $\boldsymbol{\zeta}_{\kappa-1}$, perform the following iterative step:

$$\boldsymbol{\xi}_\kappa = \boldsymbol{F}_1^{\nu_1}(\boldsymbol{\xi}_{\kappa-1}, \boldsymbol{\zeta}_{\kappa-1}) \ , \tag{47}$$

$$\boldsymbol{\zeta}_\kappa = \boldsymbol{F}_2^{\nu_2}(\boldsymbol{\zeta}_{\kappa-1}, \boldsymbol{\xi}_{\kappa-1}) \ ; \tag{48}$$

meaning that in Eq.(47) the iteration Eq.(45) has been performed ν_1 times with a fixed $\boldsymbol{\zeta}_{\kappa-1}$, and in Eq.(48) the iteration Eq.(46) has been performed ν_2 times with a fixed $\boldsymbol{\xi}_{\kappa-1}$. Very often one takes $\nu_1 = \nu_2 = 1$. The starting values $\boldsymbol{\xi}_0$ and $\boldsymbol{\zeta}_0$ are provided by the extrapolation mapping Eq.(10).

5.2 Non-linear Block-Gauss-Seidel

It is well known, that usually the corresponding Gauss-Seidel process converges faster [26], and so we are led to: Given $\boldsymbol{\xi}_{\kappa-1}$ and $\boldsymbol{\zeta}_{\kappa-1}$, do

$$\boldsymbol{\xi}_\kappa = \boldsymbol{F}_1^{\nu_1}(\boldsymbol{\xi}_{\kappa-1}, \boldsymbol{\zeta}_{\kappa-1}) \ , \tag{49}$$

and then, with the newly computed $\boldsymbol{\xi}_\kappa$, do

$$\boldsymbol{\zeta}_\kappa = \boldsymbol{F}_2^{\nu_2}(\boldsymbol{\zeta}_{\kappa-1}, \boldsymbol{\xi}_\kappa) \ . \tag{50}$$

This uses the new information as soon as it is available, and in the context of domain decomposition it would be called a multiplicative or serial Schwarz procedure [36]. Let us remark that this is very similar to the staggering variant Eqs.(12,14), only that what before was time stepping is now iteration. The starting values $\boldsymbol{\xi}_0$ and $\boldsymbol{\zeta}_0$ are again given by the extrapolation mapping Eq.(10).

Naturally, for both Eqs.(47,48) and Eqs.(49,50) we have to ask the question whether the iterative process converges. In this situation it is useful to note the following result [27, 35]:

Theorem 1 *Let*

$$\alpha = \max_{t\in[0,T]} \| \left(D_{y_2}\boldsymbol{g}_2\right)^{-1} D_z\boldsymbol{g}_2 \left(D_{y_1}\boldsymbol{h}\left(D_{y_1}\boldsymbol{g}_1\right)^{-1} D_z\boldsymbol{g}_1\right)^{-1} D_{y_2}\boldsymbol{h}\| \,, \tag{51}$$

and let L be the Lipschitz-constant of the extrapolation Eq.(10). Assume that $\alpha < 1$, and that at least κ iterations of the block-Gauss-Seidel scheme Eqs.(49,50) are performed, such that $L\alpha^\kappa < 1$, and that Δt is small enough. Then the global block-Gauss-Seidel method converges, and the global error in the n-th time step

$$\delta^{(n)} = \|\boldsymbol{x}^{(n)} - \boldsymbol{x}(t_n)\| + \|\boldsymbol{y}^{(n)} - \boldsymbol{y}(t_n)\| + \|\boldsymbol{z}^{(n)} - \boldsymbol{z}(t_n)\|$$

is bounded by

$$\delta^{(n)} \leq C(\mu^{\max\{0,\kappa-2\}}\delta^{(n)}\boldsymbol{x} + \mu^{\kappa-1}\delta^{(n)}\boldsymbol{y}) + \varepsilon_1^{(n)} + \varepsilon_2^{(n)}. \tag{52}$$

Here $\varepsilon_1^{(n)}$ and $\varepsilon_2^{(n)}$ are the errors incurred by the single system integrators Eqs.(23–25) and $\delta^{(n)}\boldsymbol{y}$ and $\delta^{(n)}\boldsymbol{z}$ are the extrapolation errors, and $\mu = \alpha + O(\Delta t)$.

If $\alpha < 1$ and Δt is small enough such that $\mu < 1$, the iteration will converge. We see that the contractivity constant α is crucial, and the convergence depends strongly on the ordering of the subsystems in the block-Gauss-Seidel solution strategy.

If enough iterations are performed, essentially only the error components from the single system integrators remain. If we have $\varepsilon_1^{(n)} = O((\Delta t)^p)$ and $\varepsilon_2^{(n)} = O((\Delta t)^q)$ as convergence orders for the single system integrators, we obtain $\delta^{(n)} = O((\Delta t)^{\min(p,q)})$, in contrast to the staggering scheme where we only have $O(\Delta t)$.

We now want to summerise for our example of FSI how the solution strategy, the grouping of the systems and algebraic constraint equations, and the order in the Gauss-Seidel process determine the contraction constant α. The computations leading to this are a bit tedious, and details will be given elsewhere.

- Assume that we have the subsystems and ordering

  1. Fluid plus coupling conditions,
  2. and the structure.

  Then the contractivity constant is

  $$\alpha = \|\boldsymbol{M}_s^{-1}\boldsymbol{T}_s^T\tilde{\boldsymbol{M}}_\tau^{-1}\boldsymbol{T}_s\| \,, \tag{53}$$

  where $\tilde{\boldsymbol{M}}_\tau$ is some sort of mass matrix acting on the coupling boundary.

- Assume the subsystems and ordering

  1. Structure plus coupling conditions,
  2. and the fluid.

  Then the contractivity constant is

  $$\alpha = \|(\ldots)\boldsymbol{M}_f^{-1}\boldsymbol{T}_f^T(\boldsymbol{T}_s\boldsymbol{M}_s^{-1}\boldsymbol{T}_s^T)^{-1}\boldsymbol{T}_f\| \,, \tag{54}$$

  where $(\ldots)$ is a lengthy but not so interesting expression.

- Assume the ordering and subsystems as

  1. The fluid, and
  2. the structure and coupling conditions.

  Then the contractivity constant is

  $$\alpha = \|(\boldsymbol{T}_s\boldsymbol{M}_s^{-1}\boldsymbol{T}_s^T)^{-1}\tilde{\boldsymbol{M}}_\tau\| \,, \tag{55}$$

  with the same $\tilde{\boldsymbol{M}}_\tau$ as in Eq.(53).

- Finally, assume the following subsystems and ordering

  1. Structure plus coupling conditions,
  2. and the fluid.

  Then the contractivity constant is

  $$\alpha = \|\tilde{\boldsymbol{M}}_\tau^{-1}\boldsymbol{T}_s\boldsymbol{M}_s^{-1}\boldsymbol{T}_s^T\| \,, \tag{56}$$

  which is the norm of the inverse of the matrix whose norm is taken in Eq.(55).

In our view this strong dependence on ordering and grouping in the Gauss-Seidel process calls for strong or tight coupling methods which will converge unconditionally provided the time step is small enough; this then would roughly be a situation similar for the single system implicit integrator. With block-Gauss-Seidel methods there is no easy way of achieving this, although there are possibilities of preconditioning [27].

5.3 Block-Newton

The possibility for such an algorithm can be sought in the group of Newton-like methods, we shall here for the sake of brevity consider the full Newton-Raphson method. It is also in this context the more challenging, as it requires the solution of a global system of linear equations in each iteration, and especially as in the partitioned approach we have no recourse to the cross-coupling parts of the system matrix. With the Newton algorithm we have a robust method, where the convergence behaviour does not depend on

- in which subsystem we include the coupling,
- and in which order we solve for the subsystems.

Usually we would start from Eqs.(23–25), the "equilibrium" equations; but it has been noted [32] that the solutions of Eqs.(23–25) are solutions of the fixed point equations Eqs.(45,46) and vice versa, but the latter are numerically much better conditioned as they implicitly already have the effect of the single system solvers in them. Hence, we perform the Newton-Raphson iteration on the fixed point equations corresponding to the iteration Eqs.(45,46), with $\Delta\boldsymbol{\xi}_\kappa := \boldsymbol{\xi}_{\kappa+1} - \boldsymbol{\xi}_\kappa$ and $\Delta\boldsymbol{\zeta}_\kappa := \boldsymbol{\zeta}_{\kappa+1} - \boldsymbol{\zeta}_\kappa$, and the iteration counter κ:

$$\begin{bmatrix} \boldsymbol{I} - D_\xi \boldsymbol{F}_1 & D_\zeta \boldsymbol{F}_1 \\ D_\xi \boldsymbol{F}_2 & \boldsymbol{I} - D_\zeta \boldsymbol{F}_2 \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\xi}_\kappa \\ \Delta\boldsymbol{\zeta}_\kappa \end{bmatrix} = - \begin{bmatrix} \boldsymbol{\xi}_\kappa - \boldsymbol{F}_1(\boldsymbol{\xi}_\kappa, \boldsymbol{\zeta}_\kappa) \\ \boldsymbol{\zeta}_\kappa - \boldsymbol{F}_2(\boldsymbol{\zeta}_\kappa, \boldsymbol{\xi}_\kappa) \end{bmatrix} . \tag{57}$$

We only want to use the existing solvers, i.e. the iteration mappings $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$. In particular, we do not have direct access to the partial derivative cross terms in the global system matrix in Eq.(57). But if we solve the system Eq.(57) by an iterative method [34, 25, 30, 31], all we need is a way to compute the product of the Jacobian matrix in Eq.(57) by an arbitrary vector, at least approximately.

6 Solving the Newton System

As we now want to consider just a single iteration, we drop the iteration index κ, and only look at computing the vector $[\Delta\boldsymbol{\xi}, \Delta\boldsymbol{\zeta}]^T$ from the right hand side. For a start, we use — symbolically — block-Gauss elimination [33, 25] on the system Eq.(57):

$$\Delta\boldsymbol{\xi} = -(\boldsymbol{I} - D_\xi \boldsymbol{F}_1)^{-1}(\boldsymbol{\xi} - \boldsymbol{F}_1(\boldsymbol{\xi}, \boldsymbol{\zeta})) - \boldsymbol{C}\Delta\boldsymbol{\zeta} \ , \tag{58}$$

with the multiplier matrix

$$\boldsymbol{C} := (\boldsymbol{I} - D_\xi \boldsymbol{F}_1)^{-1}[D_\zeta \boldsymbol{F}_1] \ . \tag{59}$$

Inserting this into the second equation, we obtain

$$\boldsymbol{S}\Delta\boldsymbol{\zeta} := (\boldsymbol{I} - [D_\zeta \boldsymbol{F}_2] - [D_\xi \boldsymbol{F}_2]\boldsymbol{C})\,\Delta\boldsymbol{\zeta} = -\boldsymbol{r}\ , \tag{60}$$

where $\boldsymbol{S}$ is the Schur complement matrix, and

$$\begin{aligned} \boldsymbol{r} &:= (\boldsymbol{\zeta} - \boldsymbol{F}_2(\boldsymbol{\zeta},\boldsymbol{\xi})) + [D_\xi \boldsymbol{F}_2]\boldsymbol{q}\ , \\ &\quad \text{where we have set} \\ \boldsymbol{q} &:= -(\boldsymbol{I} - D_\xi \boldsymbol{F}_1)^{-1}(\boldsymbol{\xi} - \boldsymbol{F}_1(\boldsymbol{\xi},\boldsymbol{\zeta}))\ . \end{aligned} \tag{61, 62}$$

Hence we solve Eq.(60) for $\Delta\boldsymbol{\zeta}$, and with this then solve Eq.(58) for $\Delta\xi$. One step of this Newton-Raphson iteration may now be formulated as follows [25, 29, 30, 31]:

1. From Eq.(62), solve

   $$(\boldsymbol{I} - D_\xi \boldsymbol{F}_1)\boldsymbol{q} = \boldsymbol{F}_1(\boldsymbol{\xi},\boldsymbol{\zeta}) - \boldsymbol{\xi} \qquad \text{for} \qquad \boldsymbol{q}\ . \tag{63}$$

2. Solve the Schur-complement system Eq.(60) $\boldsymbol{S}\Delta\boldsymbol{\zeta} = -\boldsymbol{r}$ for $\Delta\boldsymbol{\zeta}$ with

   $$\boldsymbol{r} = \boldsymbol{\zeta} - (\boldsymbol{F}_2(\boldsymbol{\zeta},\boldsymbol{\xi}) - [D_\xi \boldsymbol{F}_2]\boldsymbol{q}) \tag{64}$$

   from Eq.(61).

3. Compute $\Delta\boldsymbol{\xi} = \boldsymbol{q} - \boldsymbol{C}\Delta\boldsymbol{\zeta}$. This involves application of the multiplier matrix $\boldsymbol{C}$ from Eq.(59).

6.1 Solution of a System with a Diagonal Sub-Block

We look at the first step in Eq.(63): To solve for $\boldsymbol{q}$, we use the iterative solver for the first subsystem $\boldsymbol{F}_1$. The resolution of Eq.(63) can also be seen as one Newton-Raphson step for the solution $\boldsymbol{q}$ of the equation

$$\boldsymbol{\xi} - \boldsymbol{F}_1(\boldsymbol{\xi} + \boldsymbol{q}, \boldsymbol{\zeta}) = \boldsymbol{0} \tag{65}$$

when $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are fixed. But Eq.(65) is exactly what the iterative solver $\boldsymbol{F}_1$ is designed for. Hence with the iterative solver $\boldsymbol{F}_1$ we obtain

$$\boldsymbol{q} \approx \boldsymbol{\eta}_m - \boldsymbol{\xi}\ , \tag{66}$$

where $\boldsymbol{\eta}_m$ is the result of the iterative process

$$\boldsymbol{\eta}_{j+1} = \boldsymbol{F}_1(\boldsymbol{\eta}_j, \boldsymbol{\zeta})\ , \quad j = 0, 1, \ldots, m-1 > 0\ , \quad \text{with} \quad \boldsymbol{\eta}_0 = \boldsymbol{\xi}\ , \tag{67}$$

using the solver $\boldsymbol{F}_1$ of the first subsystem m times.

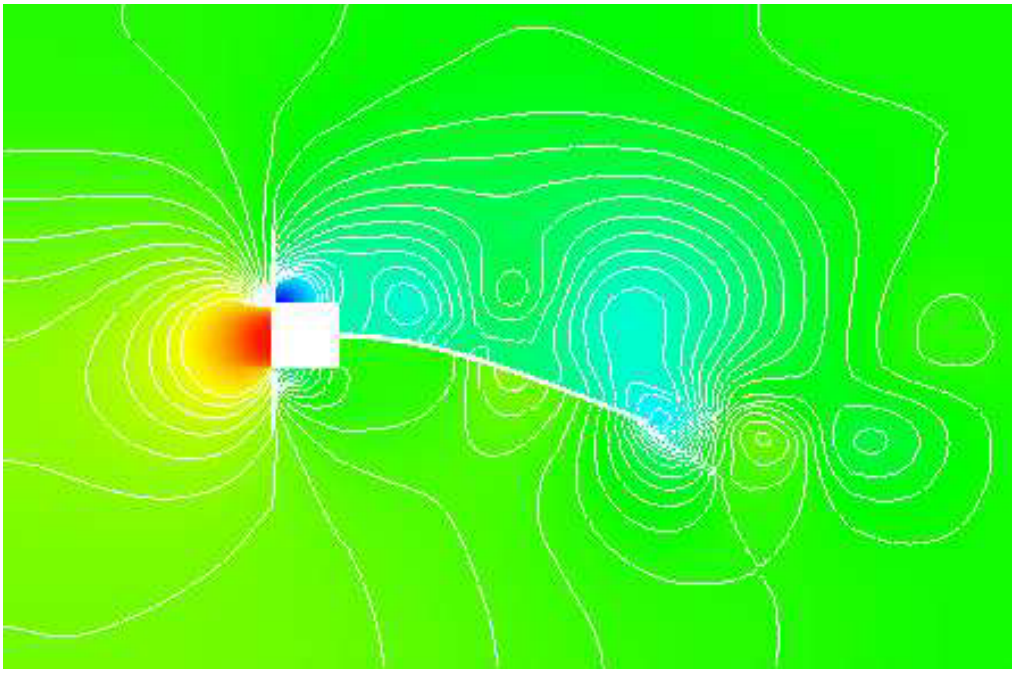

Figure 1: Vortex Shedding and Pressure Field.

6.2 Application of the Multiplier Matrix

We shall now next look at the third step, as the techniques used here are needed also in the second step, which is the most involved. In step 3 the application of the multiplier matrix $\boldsymbol{C}$ from Eq.(59) is the only point which needs some explanation. Looking at $\boldsymbol{C}$ in Eq.(59), we see that it needs the solution of a linear system with a diagonal block $(\boldsymbol{I} - D_\xi \boldsymbol{F}_1)$ — but this is exactly the same task as in the preceeding section 6.1. The other part is to apply the cross-derivative $D_\zeta \boldsymbol{F}_1$. This we do by using finite differences. For some arbitrary vector $\boldsymbol{s}$ we want $[D_\zeta \boldsymbol{F}_1]\boldsymbol{s}$, and with an appropriate small step h we have

$$[D_\zeta \boldsymbol{F}_1]\boldsymbol{s} \approx \frac{1}{h}(\boldsymbol{F}_1(\boldsymbol{\xi}, \boldsymbol{\zeta} + h\boldsymbol{s}) - \boldsymbol{F}_1(\boldsymbol{\xi}, \boldsymbol{\zeta})). \tag{68}$$

This uses the iterative solver $\boldsymbol{F}_1$ for the first subsystem once.

6.3 Solution of the Schur-Complement System

Now we can look at step 2. It involves two sub-steps: Computation of the vector $\boldsymbol{r}$ in Eqs.(61,64) and solution of the Schur-complement system Eq.(60).

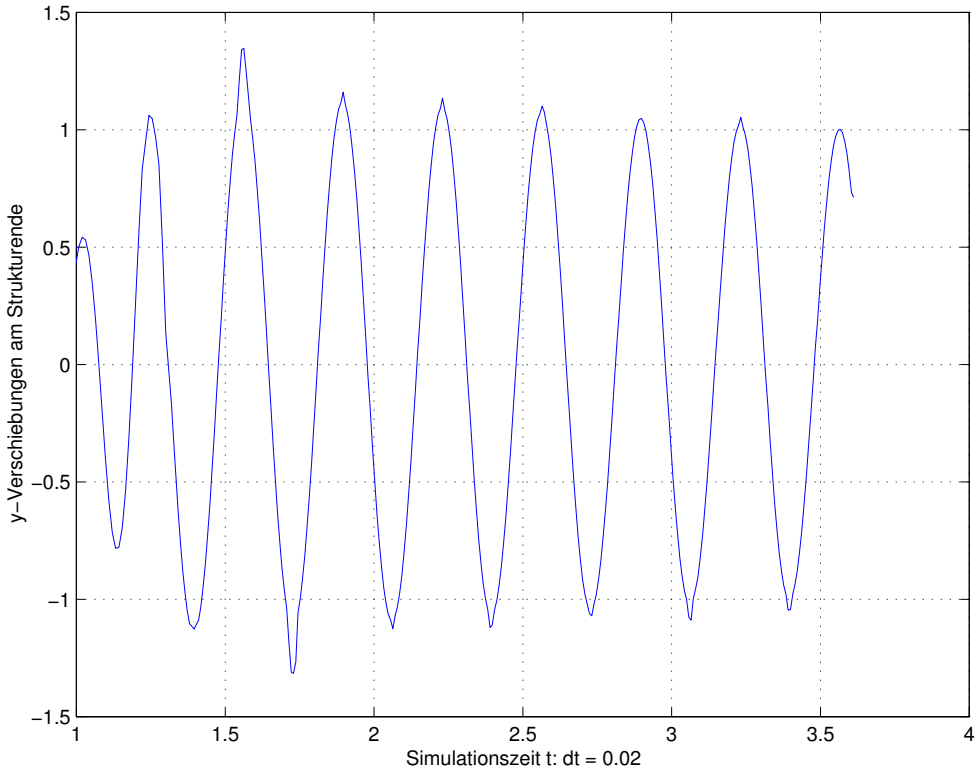

Figure 2: Displacement Response for Strong Coupling.

6.3.1 Application of an Off-Diagonal Sub-Block

To apply $[D_\xi \boldsymbol{F}_2]$ in the computation of $\boldsymbol{r}$ in Eqs.(61,64), we use finite differences as in the preceeding section 6.2, but in a reversed way. From Eq.(64), we have analogously to Eq.(68):

$$\boldsymbol{r} = \boldsymbol{\zeta} - (\boldsymbol{F}_2(\boldsymbol{\zeta}, \boldsymbol{\xi}) - [D_\xi \boldsymbol{F}_2]\boldsymbol{q}) \approx \boldsymbol{\zeta} - \boldsymbol{F}_2(\boldsymbol{\zeta}, \boldsymbol{\xi} - \boldsymbol{q}) \; . \tag{69}$$

This uses the iterative solver $\boldsymbol{F}_2$ for the second subsystem once.

6.3.2 Application of the Schur-Complement Matrix

To solve the system with the Schur complement matrix $\boldsymbol{S}$ in Eq.(60), we use an iterative Krylov method — here BiCGStab — and we only need the action of $\boldsymbol{S}$ on some arbitrary vector $\boldsymbol{s}$, again approximated via finite differences — in a reversed

way as in the preceeding sub-step — with some (small) step-size h:

$$(70)\quad \boldsymbol{S}\boldsymbol{s} = \frac{1}{h}\boldsymbol{S}(h\boldsymbol{s}) = \frac{1}{h}[(\boldsymbol{I} - D_\zeta \boldsymbol{F}_2)(h\boldsymbol{s}) - [D_\xi \boldsymbol{F}_2]\boldsymbol{C}(h\boldsymbol{s})] \approx \frac{1}{h}[h\boldsymbol{s} + \boldsymbol{F}_2(\boldsymbol{\zeta} - h\boldsymbol{s}, \boldsymbol{\xi} - \boldsymbol{C}(h\boldsymbol{s})) - \boldsymbol{F}_2(\boldsymbol{\zeta}, \boldsymbol{\xi})] \ .$$

This uses the iterative solver $\boldsymbol{F}_2$ for the second subsystem once. We need in Eq.(70) $\boldsymbol{C}(h\boldsymbol{s})$, but this we know how to compute from section 6.2.

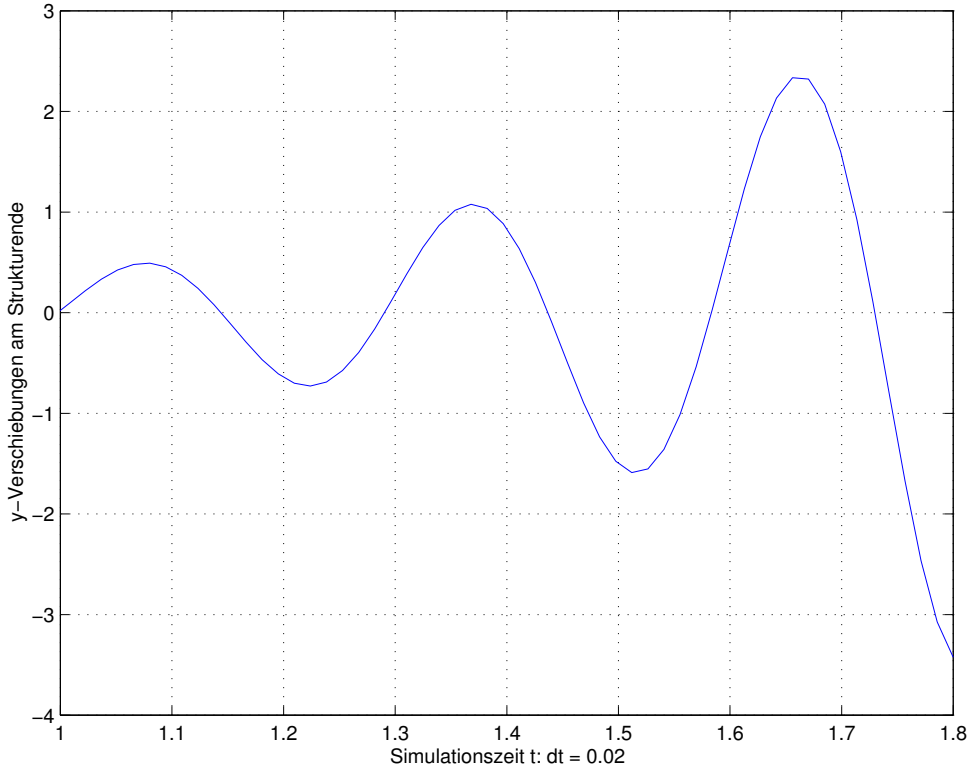

Figure 3: Displacement Response for Weak Coupling.

Now all steps for the approximative block Newton-Raphson method have been explained. Convergence of these approximative Newton methods has been investigated in [38, 39, 33, 34], and we have the following result:

Theorem 2 *If the single system solvers are quadratically convergent (or enough iterations are made in the approximative steps), the global iteration is also quadratically convergent.*

But in our view even more important than the quadratic convergence is the fact that in this way we have the same convergence characteristics as if we were using

the Newton-Raphson method in a monolithic approach, in contrast to theorem 1, where convergence in the Gauss-Seidel process depends on subsystem grouping and ordering. Naturally, the numerical work of the block Newton-method will depend on the subsystem grouping and ordering, as the subsystem solvers are not used in a symmetric way. The computationally most advantageous ordering has to be determined according to the effort involved in using each of the subsystem solvers once.

7 Numerical Examples

The methods presented so far will be demonstrated [25] on a small two-dimensional example, introduced in [43]. It is a rigid block with a thin elastic appendage, placed in an incompressible viscous flow coming from the left. The bluff shape of the square block causes separation and vortices, which are transported along the elastic appendage. Although the setup is symmetric, the vortices develop unsymmetrically, alternating from top and bottom. The pressure variations in the vortices interact with the elastic appendage, which in turn starts oscillating. This situation is shown in Fig. 1, together with the pressure field in the flow.

In Fig. 2 the vertical displacement response of the tip of the elastic appendage is shown. The computations were performed with the strong coupling algorithm introduced in sections 2 and 5.3. If we solve this problem with the same time step but with the staggering or so called weak coupling method — keeping the same subsystem solvers as before, we observe the response in Fig. 3, and one immediately recognises the instability of the staggering or weak coupling scheme.

In another test, we compare the block-Gauss-Seidel method as described in section 5.2 with the block-Newton method from section 5.3. For the same configuration as before, we show the number of iterations of either method in each time step in 4. The superior convergence characteristic of the block-Newton method is obvious.

As the block-Newton method is more expensive to perform per iteration, this does not necessarily translate into an overall performance gain — of course apart from those cases where the block-Newton converges and the block-Gauss-Seidel method does not. Therefore, as a measure of numerical expenditure, we count the number of times the subsystem solvers have to be called per time step. This is shown in 5, and again the advantage is on the block-Newton side.

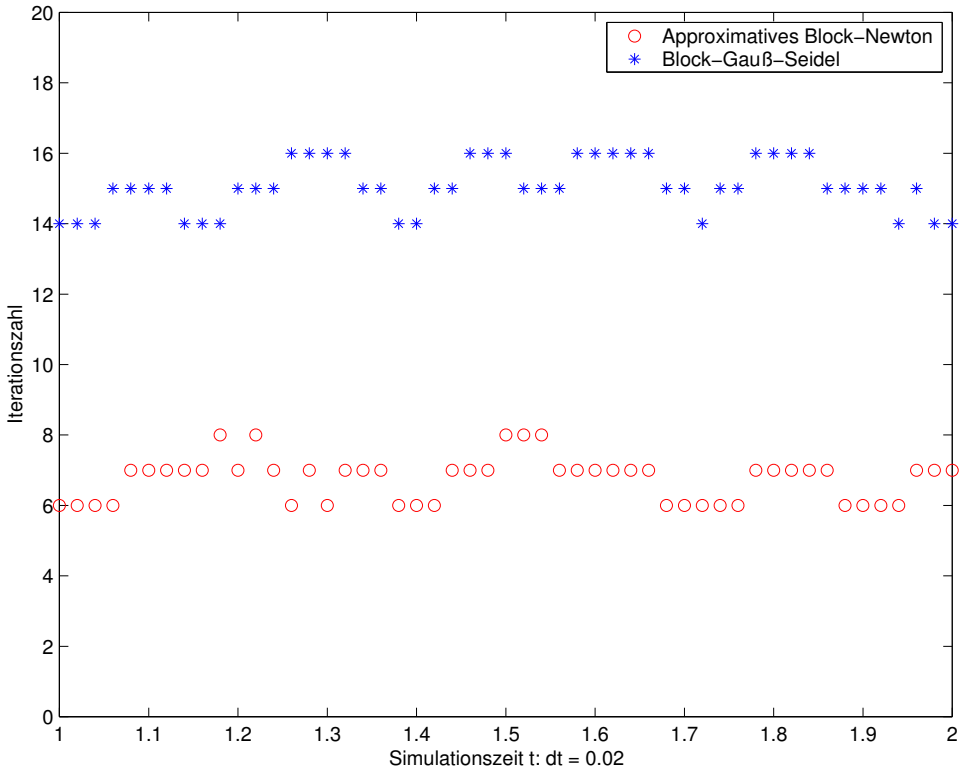

Figure 4: Iteration Count for block-Gauss-Seidel and block-Newton.

8 Conclusion

For coupled but *partitioned* problems, we have proposed a *strong* or *tight* coupling method, which achieves the same results as a monolithical approach. We have formulated this coupling problem as a differential algebraic equation (DAE) following [35, 27]. As usually appropriate to the numerical treatment of DAEs, we consider globally implicit methods.

The global system to be solved in each time step is treated with the use of the solvers for the individual subsystems. The non-linear block-Jacobi and non-linear block-Gauss-Seidel methods come very naturally, but they are sub-optimal and not robust. We have discussed the problems which arise with this approach. We propose to solve the global implicit equations with an approximate block-Newton method, following [32, 33, 34]. The linear system which arises in each iteration is solved via the subsystem solvers and a Krylov iterative method.

This iterative method only requires the possibility to use the sub-problem solvers in an iteration-by-iteration fashion. The computation of the derivatives

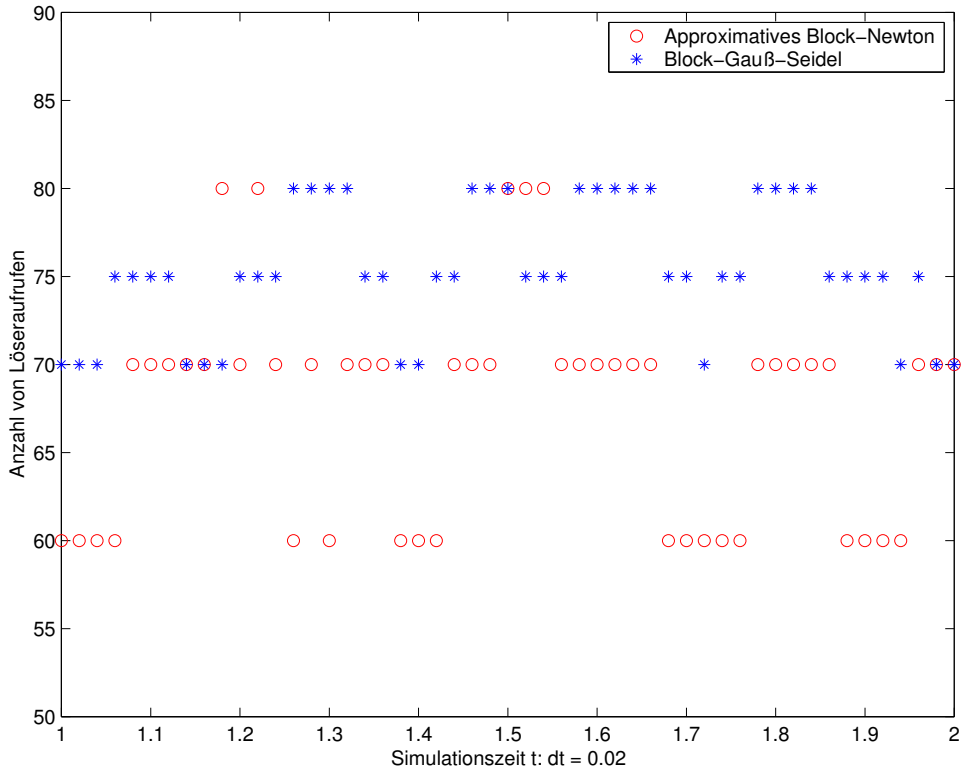

Figure 5: Count of Subsystem Solver Calls for block-Gauss-Seidel and block-Newton.

is approximated by the sub-problem solvers and by finite differencing. These methods are faster both in number of iterations and in numerical effort than the usual staggering methods; additionally they have superior convergence characteristics independent of sub-problem grouping and ordering, unlike the staggering resp. block-Gauss-Seidel method. We have demonstrated the methods on a fluid-structure interaction problem, first fitting it into the general framework, and then applying the proposed procedures. Needless to say that the general ideas about formulating and solving coupling problems are not specific to fluid-structure interaction and can be used in other circumstances as well.

The partitioned approach offers modularity both in the formulation and modeling of physical phenomena, and — a point of paramount practical importance — especially in the implementation of software [57]. This allows the use of existing software packages for the subsystems, and this was how the computations presented here were done. The structural solver was *FEAP* [58, 11], and the flow solver used was *FEATFLOW* [59, 60], which could be taken as building blocks

and only needed a small interface to be able to perform the coupling [25]. Software designed to do the clerical work of managing the communication between different software packets and their meshes [61] has been employed, and was of great help. With this kind of approach, the effort and special knowledge and techniques which have gone into designing and implementing software for specific application areas can be re-used, and the coupled simulation can benefit from this, and things do not have to be designed from scratch.

In closing, we would like to emphasise the fact that it is possible to have the above mentioned advantages of the partitioned approach, and at the same time the superior robustness and convergence characteristics of a monolithical approach — in our view this is the best of both worlds.

References

[1] H. Morand, R. Ohayon: *Fluid-Structure Interaction*. John Wiley & Sons, Chichester, 1995

[2] T. Belytschko, R. Mullen: Two-dimensional fluid-structure impact computations with regularization. Comp. Meth. Appl. Mech. Eng. **27** (1981) 139–154

[3] J. M. Kennedy, T. Belytschko: A survey of computational methods for fluid-structure analysis of reactor safety. Nuclear Engineering and Design **69** (1982) 379–398

[4] J. Donea, S. Giuliani, J. P. Halleux: An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interaction. Comp. Meth. Appl. Mech. Eng. **33** (1982) 689–723

[5] K.-J. Bathe, H. Zhang, M. H. Wang: Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions. Computers & Structures **56** (1995) 193–213

[6] S. Piperno, C. Farhat, B. Larrouturou: Partitioned procedures for the transient solution of coupled aeroelastic problems. Comp. Meth. Appl. Mech. Eng. **124** (1995) 79–112

[7] J. Mouro, P. Le Tallec: Fluid structure interaction with large structural displacements. Comp. Meth. Appl. Mech. Eng. **190** (2001) 3039–3067

[8] C. Farhat: Parallel and distributed solution of coupled nonlinear dynamic aeroelastic response. In: M. Papadrakakis (ed.) *Parallel Solution Methods in Computational Mechanics*. John Wiley & Sons, Chichester, 1997

[9] S. Rifai, Z. Johan, W.-P. Wang, J.-P. Grisval, T. J. R. Hughes, M. Ferencz: Multiphysics simulation of flow induced vibrations and aeroelasticity on parallel computing platforms. Comp. Meth. Appl. Mech. Eng. **174** (1999) 393–417

[10] K.-J. Bathe, H. Zhang, S. H. Ji: Finite element analysis of fluid flows fully coupled with structural interactions. Computers & Structures **72** (1999) 1–16

[11] O. C. Zienkiewicz, R. L. Taylor: *The Finite Element Method*. Vol. 1–3. Butterworth Heinemann, London, 2001

[12] J. Argyris, I. S. Doltsinis, P. Pimenta, H. Wüstenberg: Thermomechanical response of solids at high strains — natural approach. Comp. Meth. Appl. Mech. Eng. **32** (1982) 3–57

[13] O. C. Zienkiewicz, D. K. Paul, A. H. C. Chan: Unconditionally stable staggered solution procedure for soil-pore fluid interaction problems. Int. J. Num. Meth. Eng. **26** (1988) 1039–1055

[14] K. C. Park, C. A. Felippa: Partitioned analysis of coupled systems. In: T. Belytschko, T. J. R. Hughes (eds.) *Computational Methods in Transient Analysis*. North-Holland, Amsterdam, 1983

[15] K. C. Park, C. A. Felippa: Recent developments in coupled field analysis methods. In: R. W. Lewis, P. Bettes, E. Hinton (eds.) *Numerical Methods in Coupled Systems*. John Wiley & Sons, Chichester, 1984

[16] O. C. Zienkiewicz: Coupled problems and their numerical solution. In: R. W. Lewis, P. Bettes, E. Hinton (eds.) *Numerical Methods in Coupled Systems*. John Wiley & Sons, Chichester, 1984

[17] O. C. Zienkiewicz, A. H. C. Chan: Coupled problems and their numerical solution. In: I. S. Doltsinis (ed.) *Advances in Computational Nonlinear Mechanics*. Springer, Berlin, 1989

[18] F. J. Blom: A monolithical fluid-structure interaction algorithm applied to the piston problem. Comp. Meth. Appl. Mech. Eng. **167** (1998) 369–391

[19] S. Rugonyi, K.-J. Bathe: On the analysis of fully-coupled fluid flows with structural interactions – a coupling and condensation procedure. Int. J. of Comp. Civil and Struct. Eng. **1** (2000) 29–41

[20] C. A. Felippa, K. C. Park: Staggered transient analysis procedures for coupled mechanical systems: formulation. Comp. Meth. Appl. Mech. Eng. **24** (1980) 61–111

[21] K. C. Park: Partitioned transient analysis procedures for coupled-field problems: stability analysis. J. Appl. Mech. **47** (1980) 370–376

[22] S. Piperno: Explicit/implicit fluid-structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. Int. J. Num. Meth. Fluids **25** (1997) 1207–1226

[23] D. P. Mok, W. A. Wall: Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In: W. A. Wall, K.-U. Bletzinger, K. Schweizerhof (eds.) *Trends in Computational Structural Mechanics*. CIMNE, Barcelona, 2001

[24] J. Steindorf, H. G. Matthies: Efficient partitioned methods for the computation of fluid-structure interaction on parallel computers. In: B. H. V. Topping (ed.) *Proceedings of the Third Euro-Conference on Parallel and Distributed Computing for Computational Mechanics*. Civil-Comp Press, Edinburgh, 1999

[25] J. Steindorf: Partitionierte Verfahren für Probleme der Fluid-Struktur Wechselwirkung, PhD thesis; Technische Universität Braunschweig, 2002

[26] R. Codina, M. Cervera: Block-iterative algorithms for nonlinear coupled problems. In: M. Papadrakakis, G. Bugeda (eds.) *Advanced Computational Methods in Structural mechanics*. CIMNE, Barcelona, 1996

[27] M. Arnold, M. Günther: Preconditioned dynamic iteration for coupled differential-algebraic systems. BIT Numerical Mathematics **41** (2001) 1–25

[28] H. G. Matthies, J. Steindorf: Efficient iteration schemes for non-linear fluid-structure interaction problems. In: B. H. V. Topping (ed.) *Computational Mechanics: Techniques and Developments*. Civil-Comp Press, Edinburgh, 2000

[29] H. G. Matthies, J. Steindorf: How to make weak couplings strong. In: K.-J. Bathe (ed.) *Computational Fluid and Solid Mechanics*. Elsevier, Amsterdam, 2001

[30] H. G. Matthies, J. Steindorf: Fully coupled fluid-structure interaction using weak coupling. Proc. in Appl. Math. and Mech. **1**(1)(2002) 37–38

[31] H. G. Matthies, J. Steindorf: Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction. submitted to Computers & Structures, 2002

[32] S. Artlich, W. Mackens: Newton-coupling of fixed point iterations. In: W. Hackbusch, G. Wittum (eds.) *Numerical Treatment of Coupled Systems*. Vieweg, Braunschweig, 1995

[33] W. Mackens, J. Menck, H. Voss: Numerical system synthesis: Concepts for coupling subsystem solvers. Technical Report, Technische Universität Hamburg-Harburg, 1998

[34] W. Mackens, J. Menck, H. Voss: Numerical coupling of subsystems. Zeitschr. für Angew. Math. und Mech. **79**(S3) (1999) S871–S872

[35] M. Arnold: Constraint partitioning in dynamic iteration methods. submitted to Zeitschr. für Angew. Math. und Mech., 2000

[36] B. Smith, P. Bjorstad, W. Gropp: *Domain Decomposition*. Cambridge University Press, Cambridge, 1996

[37] K. Burrage: *Parallel and Sequential Methods for Ordinary Differential Equations*. Clarendon Press, Oxford, 1995

[38] T. F. Chan: An approximate Newton method for coupled nonlinear systems. SIAM J. Numer. Anal. **22**(5) (1985) 904–913

[39] U. Miekkala, O. Nevanlinna: An approximate Newton method for coupled nonlinear systems. SIAM J. Sci. Stat. Comput. **8** (1987) 459–482

[40] C. Farhat, M. Lesoinne: Two efficient staggered algorithms for the serial and parallel solution of three-dimensional transient aeroelastic problems. Comp. Meth. Appl. Mech. Eng. **182** (2000) 499–515

[41] C. Farhat, K. C. Park, Y. Dubois-Pelerin: An uncoditionally stable staggered algorithm for transient finite element analysis of coupled thermoelastic problems. Comp. Meth. Appl. Mech. Eng. **85** (1991) 349–365

[42] J. Donea: Arbitrary Lagrangian-Eulerian finite element methods. In: T. Belytschko, T. J. R. Hughes (eds.) *Computational Methods in Transient Analysis*. North-Holland, Amsterdam, 1983

[43] W. A. Wall, E. Ramm: Fluid-structure interaction based upon a stabilized (ALE) finite element method. In: *Proceedings of the 4th World Congress on Computational Mechanics — New Trends and Applications*. CIMNE, Barcelona, 1998

[44] K.-J. Bathe: *Finite Element Procedures*. Prentice-Hall, Englewood Cliffs, NJ, 1996

[45] J. H. Ferziger, M. Perić: *Computational Methods for Fluid Dynamics*. Springer, Berlin, 1996

[46] P. D. Thomas, C. K. Lombard: Geometric conservation law and its application to flow computations on moving grids. AIAA Journal **17**(10)(1979) 1030–1037

[47] G. Dubini, R. Pietrabissi, F. M. Montevecchi: Fluid-structure interaction in bio-fluid mechanics. Med. Eng. Phys. **17**(2) (1995) 609–617.

[48] C. Grandmont, V. Guimet, Y. Maday: Numerical analysis of some decoupling techniques for the approximation of the unsteady fluid-structure interaction. In: *Proceedings of the Second European Conference on Numerical Mathematics*. Heidelberg, 1997

[49] J. T. Batina: Unsteady Euler airfoil solutions using unstructured dynamic meshes. AIAA Journal **28**(8) (1990) 1381–1388

[50] J. R. Cebral, R. Löhner: Conservative load projection and tracking for fluid-structure problems. AIAA Journal **35**(4) (1997) 687–692

[51] C. Farhat, M. Lesoinne, P. Le Tallec: Load and motion transfer algorithms for fluid-structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. Comp. Meth. Appl. Mech. Eng. **157** (1998) 95–114

[52] N. Maman, C. Farhat: Matching fluid and structure meshes for aeroelastic computations: a parallel approach. Computers & Structures **54** (1995) 779–785

[53] B. Wohlmuth: *Discretization Methods and Iterative Solvers Based on Domain Decomposition*. Lect. Notes in Computational Science and Engrg., Springer, Berlin, 2001

[54] P. Le Tallec, S. Mani: Conservation laws for fluid-structure interactions. Technical Report CEREMADE, Université de Paris Dauphine, 1999

[55] T. Belytschko, R. Mullen: Stability of explicit-implicit mesh partitions in time integration. Int. J. Num. Meth. Eng. **12** (1978) 1575–1586

[56] C. Farhat, M. Lesoinne, N. Maman: Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation law and distributed solution. Int. J. Num. Meth. Fluids **21** (1995) 807–835

[57] C. Farhat, M. Lesoinne, P. Stern: High performance solution of three-dimensional nonlinear elastic problems via parallel parttitoned algorithms: methodolgy and preliminary results. Adv. in Engrg. Software **28** (1997) 43–61

[58] R. L. Taylor: FEAP – A Finite Element Analysis Program, User Manual Version 6.3; Dept. of Civil and Environmental Engrg., University of California at Berkeley, CA, 1998

[59] S. Turek, C. Becker: FEATFLOW User Manual Release 1.2; Institut für Angewandte Mathematik, Universität Heidelberg, 1999

[60] S. Turek: *Efficient Solvers for Incompressible Flow Problems: An Algorithmic Approach in View of Computational Aspects*. Lect. Notes in Computational Science and Engrg. Vol. 6. Springer, Berlin, 1999

[61] M. Münch: GRISSLi Kopplungsinterface, Handbuch für die Anwendungsprogrammierung. IBM High Performance Support Center, Heidelberg; Institut für Algorithmen und Wissenschaftliches Rechnen der GMD, St. Augustin; Institut für Parallele und Verteilte Höchstleistungsrechner, Universität Stuttgart, 1998

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 1998-02

1998-02	S. Petri, M. Bolz, H. Langendörfer	Transparent Migration and Rollback for Unmodified Applications in Workstation Clusters
1998-03	M. Cohrs, E. H. A. Gerbracht, W. Struckmann	DISKUS - Ein Programm zur symbolischen Diskussion reeller elementarer Funktionen
1998-04	C. Lindig	Analyse von Softwarevarianten
1998-05	Gregor Snelting, Frank Tip	Reengineering Class Hierarchies Using Concept Analysis
1998-06	Juliana Küster Filipe	On a Distributed Temporal Logic for Modular Object Systems
1998-07	J. Schönwälder, M. Bolz, S. Mertens, J. Quittek, A. Kind, J. Nicklisch	SMX - Script MIB Extensibility Protocol Version 1.0
1998-08	C. Heimann, S. Lauterbach, T. Förster	Entwurf und Implementierung eines verteilten Ansatzes zur Lösung langrechnender Optimierungsprobleme aus dem Bereich der Ingenieurwissenschaften
1999-01	A. Zeller	Yesterday, my program worked. Today, it does not. Why?
1999-02	P. Niebert	A Temporal Logic for the Specification and Verification of Distributed Behaviour
1999-03	S. Eckstein, K. Neumann	Konzeptioneller Entwurf mit der Unified Modeling Language
1999-04	T. Gehrke, A. Rensink	A Mobile Calculus with Data
2000-01	T. Kaiser, B. Fischer, W. Struckmann	The Modula-2 Proving System MOPS
2000-02	J. Saperia, J. Schönwälder	Policy-Based Enhancements to the SNMP Framework
2000-03	A. Casties	Finite-Element-Interpolation der rumlichen Dichten eines Vielteilchensystems auf ungeordneten Gittern
2000-04	J. Koslowski	A 2-dimensional view of the Chu-construction
2000-05	S. Eckstein, P. Ahlbrecht, K. Neumann	Von parametrisierten Spezifikationen zu generierten Informationssystemen: ein Anwendungsbeispiel
2000-06	F. Strauß, J. Schönwälder, M. Mertens	JAX - A Java AgentX Sub-Agent Toolkit
2000-07	F. Strauß	Advantages and Disadvantages of the Script MIB Infrastructure
2000-08	T. Gehrke, U. Goltz	High-Level Sequence Charts with Data Manipulation
2000-09	T. Firley	Regular languages as states for an abstract automaton
2001-01	K. Diethers	Tool-Based Analysis of Timed Sequence Diagrams
2002-01	R. van Glabbeek, U. Goltz	Well-behaved Flow Event Structures for Parallel Composition and Action Refinement
2002-02	J. Weimar	Translations of Cellular Automata for Efficient Simulation
2002-03	H. G. Matthies, M. Meyer	Nonlinear Galerkin Methods for the Model Reduction of Nonlinear Dynamical Systems
2002-04	H. G. Matthies, J. Steindorf	Partitioned Strong Coupling Algorithms for Fluid-Structure-Interaction
2002-05	H. G. Matthies, J. Steindorf	Partitioned but Strongly Coupled Iteration Schemes for Nonlinear Fluid-Structure Interaction
2002-06	H. G. Matthies, J. Steindorf	Strong Coupling Methods